DECEMBER 2006

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

IN THIS ISSUE

TROJAN OPTIMIZER

Combining state-of-the-art techniques such as 'spaghetti' code, Encrypting File System (EFS), object rights manipulation, reserved file names and user-mode rootkits, Trojan.Linkoptimizer manages gracefully to avoid detection by many AV engines – and its removal can be a real nightmare. Mircea Ciubotariu has the details.

page 6

WHAT'S THE FUSS?

Jeffrey Aboud investigates the buzz surrounding trojan crimeware.

page 11

COMPARATIVE REVIEW

A diverse range of products was submitted for this month's 64-bit comparative review. John Hawes has the details of how they all fared.

page 14

vbSpam supplement

This month: anti-spam news and events; and Catalin Cosoi provides details of a technique *BitDefender* researchers have developed for dealing with image spam.

ISSN 1749-7027

'As security companies we must provide multiple layers of defence to protect our users properly.'

Robert Sandilands, Authentium

WHAT IS ANTI-VIRUS SOFTWARE?

Towards the end of 2007 you will find that anti-virus is no longer software that 'just' detects viruses. As a result of the changes in computers and their purpose, anti-virus programs have evolved into complex pieces of software that have multiple functions and protect users through a variety of techniques.

In the past, most pieces of malware were badly written and full of bugs and their effects could easily be identified by the average user. But malware writers are increasingly becoming very professional, with viruses being written on demand for specific purposes – such as stealing your money, stealing your identity or using your machine as a spam-sending zombie.

Many of these pieces of custom-written malware seem to have gone through some form of quality control process and seem to be well managed. The malware also uses a variety of different techniques and components. The components are often self-updating and protect themselves from being detected and/or removed.

One of the basic principles of computer security is layered defence. One should never depend on a single layer of defence because once that layer is breached it leaves you defenceless. The average modern piece of malware will disable security software as one of its first actions, and once the computer's security has been bypassed you don't get any second chances.

As security companies we must provide multiple layers of defence to protect our users properly. Different layers of defence can include a number of technologies: known-virus scanners, heuristics, host intrusion detection, behavioural blocking or detection, policies (both machine and human-based), reputation-based systems and firewalls. None of these technologies can provide complete protection on its own, but used together they form a good, multi-layered package to maximize the user's security.

Known-virus scanners use a variety of techniques to identify known risks. However, malware authors can use several methods to obscure viruses from scanners, with varying levels of success. Heuristic detection uses a combination of the techniques used by the known-virus scanners with some other tricks to determine the likelihood that a specific executable is a threat.

This is where the additional layers of defence prove their worth. The extra levels of protection can mean the difference between making life easy for the criminals and having a secure machine.

Unfortunately, some of these other techniques can affect the user's privacy. The products can report data about the user's habits and the actions of the security software to a central database for use in isolating threats or providing statistics on the size of the threat. Some vendors go to significant lengths to protect the user's privacy, but unfortunately this cannot be generalized.

Other technologies are invasive in a different way. They need to be able to monitor and control the actions taken by the operating system and, effectively, the user. The security software needs to become the watcher that watches the watcher. This is very complex technology that takes security to a new level, as the security software needs to understand the intent of the operating system or user, as well as what he or she is doing. These technologies need very deep access to what your computer is doing and how it is working – indeed this has become one of the sources of debate around *Microsoft*'s *Patchguard* kernel protection technology.

The anti-virus industry needs to and will continue innovating to keep users as safe as technology can make them. Sometimes the environment in which the anti-virus industry has to operate makes this task more complex than it perhaps needs to be. Despite that, the competition that exists in the anti-virus industry will ensure that customers receive innovative products that provide them with the level of security they demand.

NEWS

FESTIVE GREETINGS

The *VB* team wishes all *Virus Bulletin* readers a very happy Christmas and a prosperous and peaceful new year.

Seasons greetings from Helen and John.

This year, continuing the tradition of its Christmas charity donations, *VB* has made a donation to *Crisis*, a UK-based charity providing services and support for the homeless (http://www.crisis.org.uk/).

IN THE PICTURE?

Were you at VB94 in Jersey? *VB* has unearthed some photos taken at the fourth Virus Bulletin Conference. To see those who were the fresh faces of the anti-virus industry 12 years ago, or simply to reminisce, visit http://www.virusbtn.com/conference/vb94/photos.xml.

STOCKING FILLER

If you're stuck for a last-minute Christmas gift idea, Mike Berry's new book could be the answer. Mike Berry is the creator of 'scam-baiting' website www.419eater.com, which records his (and others') attempts to fight back at the perpetrators of 419 scams. Berry has been scam baiting for several years – replying to scammers' emails, expressing an interest in their propositions and fooling them into carrying out a variety of time-wasting and humiliating acts. Now, he has compiled a book, *Greetings in Jesus name!,* which contains the email correspondence from just a small number of his successful baiting attempts.

The book starts with a brief introduction to the 419 (a.k.a. advance fee fraud) scam, which is followed by ten chapters, each following a different baiting attempt from the receipt of the initial scam email through to its conclusion. The stories include tales of fraudsters who were coerced into carving a wooden replica of a Commodore computer, writing out by hand an entire *Harry Potter* novel, flying to Glasgow for a fictitious meeting, and even tattooing themselves with the words 'Baited by Shivers' (Shiver Metimbers being Berry's screen name).

While a lot of the stories leave one questioning the ethics of this type of activity, and the book may not be suitable reading for those who are easily offended, Berry urges 'don't be inveigled into feeling sympathy for any of the scammers in this book ... There are innumerable stories of the greed of the 419 scammers, and of their heartlessness.' So if you can ignore the prickle of your conscience and intead keep in mind the amount of damage 419 scammers cause their victims, the book will make for an entertaining read. For details see http://www.harbourbooks.co.uk/titles_detail_1.asp?ID=13.

Prevalence Table – October 2006

Virus	Type	Incidents	Reports
W32/Mytob	File	3,507,477	28.67%
W32/Netsky	File	3,337,363	27.28%
W32/Bagle	File	2,413,696	19.73%
W32/MyWife	File	1,048,023	8.57%
W32/Zafi	File	441,357	3.61%
W32/Mydoom	File	402,981	3.29%
W32/Lovgate	File	372,897	3.05%
W32/Bagz	File	357,269	2.92%
W32/Parite	File	74,444	0.61%
W32/Stration	File	43,793	0.36%
W32/Tenga	File	30,716	0.25%
W32/Mabutu	File	27,704	0.23%
W32/Klez	File	26,011	0.21%
W32/Funlove	File	24,350	0.20%
W32/Elkern	File	20,861	0.17%
W32/Valla	File	11,850	0.10%
W32/Reatle	File	10,211	0.08%
W32/Bugbear	File	9,000	0.07%
VBS/Redlof	Script	8,987	0.07%
W32/Maslan	File	8,046	0.07%
W32/Agobot	File	7,996	0.07%
W32/Sober	File	7,529	0.06%
W32/Lovelorn	File	6,156	0.05%
W32/Dumaru	File	4,892	0.04%
W32/Sality	File	3,713	0.03%
JS/Kak	Script	3,676	0.03%
W32/Plexus	File	2,005	0.02%
W32/Gurong	File	1,812	0.01%
W97M/Thus	Macro	1,593	0.01%
W32/Rontokbro	File	1,482	0.01%
W32/Chir	File	1,366	0.01%
W95/Tenrobot	File	1,117	0.01%
Others[1]		12,955	0.11%
Total		12,233,328	100%

[1]The Prevalence Table includes a total of 12,955 reports across 60 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

LEAPS AND BOUNDS

Peter Ferrie
Symantec Security Response, USA

Imagine you're a virus writer, someone who specialises in one-of-a-kind viruses, and you want to do something really new and different. You want it to be entrypoint-obscuring, using a technique that no one has used before. You want a polymorphic decryptor, one that appears to be deceptively simple. Of course, you also want a 32-bit and a 64-bit version. What would it look like? The answer is W32/Bounds and W64/Bounds!AMD64.

THE IMPORT/EXPORT BUSINESS

Bounds uses an entrypoint-obscuring technique that no one has used before. The secret lies in the Bound Import Table (hence the name of the virus), but we need to start with the Import Table.

The Import Table begins with an array of Import Directory Tables, which describe the rest of the import information. Each Import Directory Table contains the name of the DLL from which functions will be imported, the time/date stamp of the DLL, an array of function names to import, and an array of host memory locations in which to store the function addresses.

BOUND IMPORT TABLE

The Bound Import Table works in conjunction with the Import Table, and can decrease loading time for some applications.

The idea is that the array of host memory locations can be filled in advance, given the knowledge of the DLL from which functions will be imported. The assumption is that for any given DLL, the combination of its name and its time/date stamp is unique. Thus, the functions inside that DLL will always have the same addresses, and any application that uses those functions can have those addresses stored in the Import Table.

However, not all DLLs are suitable for this kind of manipulation, which brings us to the Bound Import Table. The Bound Import Table is an array of DLL names and time/date stamps for the DLLs for which the addresses are considered permanent. When the operating system loads an application, it checks whether the application contains a Bound Import Table. If it does, then the operating system checks that each time/date stamp in the Bound Import Table matches the time/date stamp for each DLL that is named in the Import Table.

If the time/date stamp matches, then the addresses that correspond to the Import Table entry for that DLL are assumed to be correct, and are not updated. If a time/date stamp does not match, or there is no entry for it in the Bound Import Table when compared to the Import Table, the address will be fetched from the DLL that is named in the Import Table in the usual way.

BOUNDARY CONDITIONS

Now let us return to Bounds. The virus appears to be based on a member of the Chiton family. Indeed, we can see from a text string in the virus body that the author is the same.

The virus behaves in much the same way as several viruses we have seen previously. It begins by retrieving the address of kernel32.dll, using the address of the ExitProcess() API as a hint to where in memory to begin looking. After gaining access to kernel32.dll, the virus will retrieve the addresses of the API functions that it requires, using the CRC method to match the names, so no strings are visible in the code. The virus then searches for files in the current directory and all subdirectories.

Files are examined for their potential to be infected, regardless of their suffix, and will be infected if they conform to a very strict set of conditions.

The first of these is that the file is not protected by the System File Checker. The remaining filters include the condition that the file being examined must be a character mode or GUI application, that the file must have no digital certificates, and that it must have no bytes outside the image. The virus also requires a particular CPU, depending on the variant of the virus. For the W32 version, the required CPU is an *Intel 386+*; for the W64 version, the required CPU is an *AMD64* or *Intel EM64T*.

ENTRYPOINT OBSCURING

The virus's entrypoint-obscuring technique works by checking first if a file has a Bound Import Table. The virus does not create its own Bound Import Table, so if a file does not have one, it will not be a candidate for infection.

If the file does have a Bound Import Table, then the virus checks whether it contains an entry for kernel32.dll. The reason is that the virus wants to hook the ExitProcess() API within the Import Table, which is exported by kernel32.dll. Thus, if kernel32.dll is not referenced by the Bound Import Table, then even if ExitProcess appears in the Import Table, its address will be replaced by the operating system whenever the application loads.

If the Bound Import Table does have an entry for kernel32.dll, then the virus searches the Import Table for the Import Directory Table that corresponds to kernel32.dll. The virus examines only the first entry that refers to kernel32.dll, since this covers the most common case. (There may be more than one entry for any given DLL, and compilers such as *Borland Delphi* produce such files, but these are exceptions.)

Once the Import Directory Table that corresponds to kernel32.dll has been found, the virus searches within the array of host memory locations for a reference to the address of the ExitProcess() API. If the address is found, it is replaced within the array by the entrypoint of the virus.

When a file that meets the infection criteria is found, it will be infected. If relocation data exists at the end of the file, the virus will move the data to a larger offset in the file, placing its own code in the gap that has been created. If there is no relocation data at the end of the file, the virus code will simply be placed here.

POLYMORPHISM

The polymorphic decryptor in Bounds is perhaps the most interesting thing about the virus. In a typical decryptor, the CPU registers are initialized to fixed values, using any combination of MOV/XOR/PUSH+POP, after which the values might be altered in obscure ways to other values.

Bounds, on the other hand, uses no such instructions to initialize the registers. Instead, only two operations are used: AND and OR. These operations are used repeatedly to initialize the individual bits within each register.

In addition to these operations, the decryptor uses the rest of the set – ADC/ADD/SBB/SUB/XOR/CMP – to obfuscate the values temporarily. Once the registers have been initialized completely, these other operations are used to alter the values permanently. The use of ADC and SBB is not random – the virus keeps track of the carry flag status, so the effects of the ADC and SBB are known.

The result is something that looks like this (W32 version):

```
81 E5 59 E6 5A ED    and  ebp, 0ED5AE659h
81 D4 0A A1 DA F9    adc  esp, 0F9DAA10Ah
81 F1 D8 AF FF 07    xor  ecx, 007FFAFD8h
81 CE A2 46 3E CB    or   esi, 0CB3E46A2h
```

or this (W64 version):

```
48 81 CE 0E EB 43 23  or   rsi, 2343EB0Eh
48 81 F0 3D DD 81 52  xor  rax, 5281DD3Dh
48 81 D4 F4 BE 9A 43  adc  rsp, 439ABEF4h
48 81 CB 36 F7 90 42  or   rbx, 4290F736h
```

An impenetrable list of instructions, all the same length.

The virus generates a random number of these instructions before it generates the real decryptor instructions. Since the two are indistinguishable, the problem is knowing where to start.

The reason this is a problem is because the ESP register is similarly transformed. Since the register values are not known to the virus prior to initializing them in the decryptor, an anti-virus CPU emulator could simply start emulating from the first instruction and eventually reach the real entrypoint of the virus. At that point, the decryptor would start to initialize the registers in the usual manner, and it would work regardless of the initial values.

Normally, this would defeat the entrypoint obscuring technique. However, the use of the ESP register means that emulating from the first instruction will result in a value of the ESP register which has been transformed in an unpredictable way. This appears to be intentional, since the decryptor then writes decrypted values to the stack prior to placing them into executable memory.

If the ESP register has been randomized, then when the decryptor starts to write to the stack as shown below, the memory location that will be touched is no longer known to be the stack.

W32 version:

```
89 84 B4 D7 7C 94 1B  mov  [esp+esi*4+1B947CD7h], eax
```

W64 version:

```
89 B4 54 7C E9 AA 84  mov  [rsp+rdx*2-7B551684h], esi
```

If the memory location happens to point instead to the decryptor code, then the decryptor will be damaged, and the virus will not work in the emulated environment.

Even without that complication, a typical decryptor will write to memory in a linear manner, so an emulator could simply find the first memory reference, then start emulating from there, knowing what value will come next in memory, and eventually recovering all of the registers to decrypt the entire code. The author of Bounds was probably aware of this. While the writes to the stack memory are linear, the values that are written there do not correspond to linear addresses within the virus code. Instead, the virus writes a random number of values to the stack, then begins to pop some of them into the virus body, as shown below.

W32 version:

```
8F 84 FD D7 29 AF 2D  pop dword ptr
[ebp+edi*8+2DAF29D7h]
```

W64 version:

```
66 8F 05 4A B4 FF FF  pop word  ptr [rip-00004BB6h]
```

The 32-bit version uses the registers to decode to a random address located earlier than the current position, not exactly at the start of the decryptor.

The 64-bit version uses RIP-relative addressing to overwrite the decryptor from the initial address. The reason for the RIP-relative addressing has to do with a limitation of register assignment: 64-bit CPUs do not support 64-bit immediate values. Therefore, the virus cannot perform 64-bit arithmetic to set the 64-bit CPU registers to point to the memory address of the decryptor.

The entire virus is never stored on the stack all at once – some values are placed onto the stack, and some values are then removed from the stack. Sometimes, more values can be written to the stack before all values are removed from the stack; sometimes all values are removed from the stack before more values are written to the stack.

OOPS

Every value in the virus is decoded individually using this method, resulting in very large decryptors. Since the size of the decryptor is hard to guess, it is easy to understand how a miscalculation could creep into the virus code.

Sure enough, while the virus always allocates enough bytes to hold the decryptor, a bug sometimes results in not all of the bytes being copied into the host. Both the 32-bit and 64-bit versions are affected, but in the case of the 64-bit version, the decryptor almost always ends before the cutoff point, so the bug is not so obvious.

CONCLUSION

So imagine that you're a virus writer, someone who specialises in one-of-a-kind viruses, and you want to do something that's really new and different. What should it be? How about quitting?

W32/Bounds, W64/Bounds!AMD64

Type:	Direct-action parasitic appender/inserter.
Size:	246kb (W32), 583kb (W64).
Infects:	*Windows* PE files (32-bit for W32, 64-bit AMD64 for W64).
Payload:	None.
Removal:	Delete infected files and restore them from backup.

TROJAN ANALYSIS

WHAT NEXT? TROJAN.LINKOPTIMIZER

Mircea Ciubotariu
Symantec Security Response, Ireland

Also known as Gromozon, the Linkoptimizer trojan has created havoc within the AV industry. It has raised alarms signalling that there are other ingenious ways besides advanced rootkits to make the lives of both users and security providers a nightmare. Combining state-of-the-art techniques such as 'spaghetti' code, Encrypting File System (EFS), object rights manipulation, reserved file names and user-mode rootkits, the trojan manages gracefully to avoid detection by many AV engines and its removal can be a real nightmare.

THE TROJAN

Trojan.Linkoptimizer has pushed the limits of persistence and stealth to a point where it manages to evade AV detection most of the time.

Recently we have seen what a significant impact advanced rootkits can have on the AV industry, but in order to achieve a really good rootkit one has to go deeper into the system, making obscure undocumented changes, therefore introducing a greater risk of system instability. This means that such techniques can be applied only to a limited number of targets, and updates are required even for slight changes in the environment.

There are other ways to make code 'stealthy', some of which have already been discussed in various articles. But so far Linkoptimizer is the first to combine these techniques and add its own flavour to them by developing new ones.

Although there is a lot to be said about this complex threat, this article will focus on the big picture of the trojan and the new elements it brings to the scene, especially its methods of evasion, how they work and how its authors have adapted them.

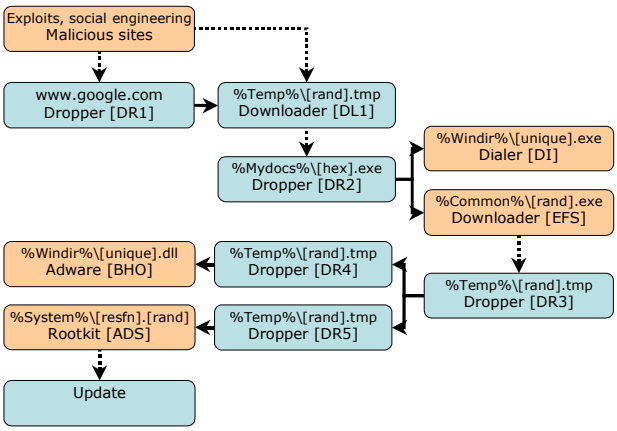
Linkoptimizer is a trojan – in fact it is an army of trojans, consisting mainly of droppers and downloaders with two ultimate purposes:

- To display advertisements.
- To dial premium-rate numbers.

The former has been identified as the sole purpose of the trojan by various vendors, with the latter somehow having been neglected as it was not associated with the 'adware' component. This might be due to the fact that the two components responsible for these actions are dropped at

different levels in the Linkoptimizer scheme. The adware component receives more attention since it is closer to the rootkit and EFS components.

The domain gromozon.com is considered to be responsible for spreading this trojan, which also gives the initial name for the threat, Gromozon. Its registration date goes back to mid February 2006, so it is expected that the first variants were released as early as March. As only some of its components were submitted individually for analysis – mainly by retail users – it was only in June/July that it came to the attention of the AV vendors and it took a while for all the pieces to be put together to get the full picture (see Figure 1).



Component	Size (KB)	Packed	Spaghetti	Naming	Auto delete	Type
DR1	10 - 17	FSG, -	Yes	Social engineering	Yes	Exe
DL1	9 - 13	-	Yes	Temporary	Yes	DLL/ OCX
DR2	61 - 67	UPX	Yes	6/8 random hex digits	Yes	Exe
DI (Dialer)	16 - 19	UPX, SUE	No	Generated machine unique	No	Exe
EFS	56 - 75	-	Yes	Reserved file names, other	No	Exe
DR3	156 - 193	UPX	Yes	Temporary	Yes	Exe
DR4	75 - 102	-	No	Temporary	Yes	Exe
BHO (Adware)	64 - 82	UPX+SUE	No	Generated machine unique	No	DLL/ BHO
DR5	126 - 153	-	Yes	Temporary	Yes	Exe
ADS (Rootkit)	115 - 139	-	Yes	Reserved file names, other	No	DLL

Figure 1: Trojan.Linkoptimizer scheme and table.

THE ACTION

As shown in Figure 1 the trojan gets into the victim's computer while browsing malicious sites. So far it is known that these sites are only using old exploits (i.e. no zero-day exploits have been used) with a little social engineering. This would make it hard to infect users with up-to-date patches and updates. However, two infection techniques have already been detailed [1, 2] and for the purpose of this paper, we will focus on the big picture.

The trojan is set in motion once the first downloader, DL1, is executed on the computer through one of the two ways shown: it may be dropped and saved as www.google.com, or another seemingly innocuous name, or installed directly as an ActiveX Control (OCX file type) object, under the name 'FreeAccess.ocx'. Once run, DL1 will attempt to download the following encrypted file:

```
shiptrop.com/1/pic.gif?<id_dec1>&<id_hex>&0
```

After decryption an executable is dropped into the current user's 'My Documents' folder. The name of the executable is made up of six or eight random hexadecimal digits (e.g. 3e22c2d.exe); this is the second dropper in the scheme, DR2. It will drop two other components: the dialer DI, and the EFS downloader.

The dialer executable is dropped into either the %Windir% or the %Windir%\Temp folder with the name generated pseudo-randomly so that it looks random, but it will always be the same on the same computer.

The dialer carries an .xml file that contains a long list of accounts, passwords and telephone numbers to be used when an active modem connection is detected in the system. Most of them are Italian premium-rate numbers starting with '899', but there are also some entries that use the *Globalstar* mobile satellite service, starting with '008819'.

Considering the authors decided to add the *Globalstar* satellite numbers just in case the compromised computer was outside Italy, it is worth noting that the number of countries from which these numbers can be dialed is limited because of the use of the prefix '00'. For example, from the USA and Canada the prefix '011' must be used to dial international numbers, so the prefix '00' will not work.

The EFS component will be discussed later, its main purpose being to download, decrypt and execute the following file:

```
shiptrop.com/2/pic2.gif?<id_dec2>& <id_hex>&0
```

This file is an intermediary dropper, DR3, which will drop two other droppers that we call DR4 and DR5. DR5 drops the rootkit component which is presented later on.

DR4 drops a DLL into the %Windir% folder. The name is similar to the dialer component and will be registered as a

Browser Helper Object (BHO) using a pseudo-randomly generated CLSID that will remain the same for the same computer, so that it will be loaded automatically by *Internet Explorer*. The purpose of the BHO component is to display advertisements retrieved through this location:

```
wlow.net/common/hint_js.php?e=<request_enc64>
```

For quite a while the trojan created an uninstall entry called 'Linkoptimizer' so that users could actually make use of the Add/Remove Programs function from the Control Panel, but the results would not be as expected. The following command would be executed:

```
iexplore.exe "http://notetol.com/uninstall.php"
```

This will render a page with only one small button in the middle reading 'Uninstall'. When the button is clicked it is replaced with a text box reading 'Thank you'. Needless to say, the threat is not touched at all, this being more of a practical joke meant to set a challenge in removing Linkoptimizer.

Recently, however, the trojan's authors have decided to change the uninstall entry from 'Linkoptimizer' to a CLSID with various names, but still the same action.

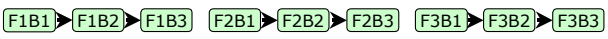
Upon execution, all of the components of the trojan attempt to detect if the running environment is a virtual machine using the 'red pill' method [3] and also check for the presence of the kernel mode debugger *SoftICE*. If either is detected they will simply exit.

SPAGHETTI CODE

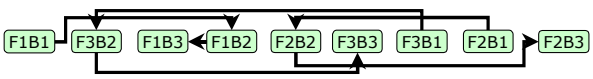
A common feature of Linkoptimizer is that it makes heavy use of spaghetti code [4] in its binaries. This is most likely where the preferred name of the trojan came from: an intermediary layer between the compiler and the linker that splits the code into many small blocks, then shuffles them and finally binds them together using jump instructions.

For example, let's consider a compiled binary having three consecutive functions, F1, F2 and F3, each of which is split into three basic code blocks, B1, B2 and B3.

The basic layout of the code right after compilation will be:



And one possible arrangement after the intermediary 'optimizer' layer might be:



From the source level – the creators' point of view – everything looks straightforward and meaningful, but what happens from the researcher's point of view? That's right, they only have the 'messy' code to deal with.

Previously, we have seen attempts to obfuscate code mainly at the source level by introducing extra garbage code – in most cases using macros that would generate different code each time and therefore different binaries. However Linkoptimizer has taken this a step further and mixed up the code with the following direct consequences:

- Analysis of such code is much slower since it is more difficult to follow.
- Automatic recognition of standard library functions (e.g. IDA FLIRT signatures), as well as the recognition of the common code between different variants of the threat is no longer possible without further time-consuming processing.
- Detection signatures on this kind of code are not efficient at all; in most cases the detection would be limited to a single file only.

As will come as no surprise, this is combined with high-frequency updates of its components – about two per week – which ensures that the newly created binaries look different each time, and therefore they won't be detectable with regular definitions.

To make analysis of such code even more frustrating, the trojan's authors have used two types of jump instructions in the binding process of the code blocks: direct jump instructions (opcode 0xe9/0xeb – jmp imm32/imm8) and indirect memory jumps (opcode 0xff, 0x25 – jmp [mem32]) where the target code block address is encoded in the data area.

All the spaghetti optimized binaries have their strings encrypted with RC4, using only one byte derived from the length of the string as the key. Decryption is performed only when the string is needed and only on the stack so that a memory dump of the module would not leak any useful information; and of course, each time the binary is compiled the string keys are changed.

Also, with the exception of a minimal set of APIs, the imports of such binaries are in some cases encrypted with RC4 as well. In other cases they are not stored at all in the file, in which case CRC32 hashes of the API names are stored.

RESERVED FILE NAMES

In order to maintain compatibility with certain DOS features, *Windows* operating systems reserve a number of file names which are used to access various physical devices. These are as follows:

Device file name	Description
AUX	Auxiliary device
COM1, COM2, …, COM9	Serial ports
CON	Console device
LPT1, LPT2, …, LPT9	Parallel ports
NUL	Null device
PRN	Printer device

Normally these file names cannot be used in conjunction with any extension, for example C:\LPT3.X would still refer to LPT3. However, 'LPT3X' is a valid file name and can be used to store data as any other file.

However with *Windows NT*, due to the limitations in the maximum path length, a new way of accessing files has been added, namely to use the '\\?\' prefix with a fully qualified file path. This not only increased the maximum allowed length of a path, but also allowed the use of the reserved device names in file or even folder names.

Two Linkoptimizer components, namely the EFS and the rootkit components, sometimes make use of this feature upon installation when naming their executables. The purpose is obvious: since most applications use the standard path names, they will be denied access to these files, therefore preventing curious eyes from seeing the contents of the files.

For example, the rootkit component may be dropped with the following name: '\\?\C:\Windows\System32\LPT1.IPJ', which will be treated as LPT1 unless the '\\?\'prefix is used.

ENCRYPTING FILE SYSTEM

Linkoptimizer effectively uses Windows Encrypting File System (EFS) [5], shipped starting with *Windows 2000*, as an anti-antivirus technique. In doing so it creates a new administrator-equivalent account with a random name and random password. Then it copies itself with some garbage data appended into one of these locations:

- %COMMONPROGRAMFILES%\System
- %COMMONPROGRAMFILES%\Services
- %COMMONPROGRAMFILES%\Microsoft Shared
- %PROGRAMFILES%\Windows NT

Next, it encrypts the file through EFS and adds a new system service with a random name pointing to it, under the credentials of the newly created user. The service is instructed to run at boot time under the new account, so that no other common user or object has access to its contents.

Moreover, it manipulates the discretionary access control list (DACL) of the registry subkey that holds the service's details, as well as the executable's DACL, to permit access to these two objects only to its owner, which is set to be the new user. This means that if anyone – even the administrator – attempts to read or change the settings of the service or attempts to delete the file, he/she will be denied access.

However, there are means of getting access to the contents of the protected objects. In the case of registry subkeys, the system built-in account has access by default to any key of the hive. This means that, using the system credentials, one could read any protected information from the registry.

With the encrypted file, however, things are a bit different. One would need to make some changes in the system in order to gain access to the decrypted contents of the file. In our case an easy solution would be to set the service's executable name to a different application that will run instead of the actual EFS and that will copy the contents of the EFS object to a new unencrypted container. At the end the service executable can be restored.

The implications of this are great: the biggest concern is not privacy invasion, since AV products may ask for permission before doing so, but access to the malicious contents, because there is no general way to get the contents of any EFS object.

The EFS component of Linkoptimizer goes a step further in preventing access to the file through the service's process, which contains the credentials of the user that it was run as. In doing so the service only runs for a small amount of time, just enough to create a new process, normally svchost.exe, inject its code into this process and perform all the subsequent actions with the appearance of an innocent and legitimate service.

Even for the small amount of time that the service runs, yet another protection measure has been introduced, namely to remove the SeDebugPrivilege from the administrator-equivalent built-in security group, which prevents any user in the system from manipulating processes or threads other than its own.

A consequence of this is that, without regaining SeDebugPrivilege, utility tools such as *Filemon* or *Regmon* will no longer run and some monitoring processes and information utilities will fail to work properly.

THE ROOTKIT

The rootkit component of Linkoptimizer is what attracted the most attention from the AV industry, although it does not bring anything new besides a certain persistence in staying up and running.

First, it is a user-mode rootkit that injects its own DLL into all running processes and hooks via patching 100 APIs from five libraries of the latest version. Initially the hooks were

intended to cloak the rootkit and the BHO component, but over time protection mechanisms have been added.

One example is the blocking of known security tools. This is done in two steps. Firstly, by checking the name of the executable that is about to be run against a blacklist; and secondly, by checking the version information strings located in the executable's resources against another blacklist. When a string is found in the blacklist the execution request will simply be ignored.

This way it effectively blocks many rootkit detectors and some fix tools provided by AV companies, which otherwise could remove or reveal some of its components.

It is noteworthy that the rootkit's cloak does not cover the dialer component. For some reason it was left outside, making it the most vulnerable component to detection and removal.

At the beginning the rootkit DLL was hidden in an Alternate Data Stream (ADS) of one of the following folders:

- System root folder (e.g. C:\:kzpy.qmt)
- Windows folder (e.g. C:\Windows:hxlga.rbs)
- System folder (e.g. C:\Windows\System32:jicqr.nvd)

In time, the shelter provided by the ADS was no longer good enough since, although Alternate Data Streams can be used as normal files – can be executed, read and written – they don't have the same properties as normal files – for instance they don't have a security descriptor, a feature that would make them easier to remove.

In the end, the authors gave up using ADS for the rootkit storage and instead they started to use reserved file names or existing file names with a couple of characters changed. This time, however, they were able to use other tricks such as manipulating the file's security descriptor to allow execute permission only.

To ensure that the rootkit runs each time the system is restarted, it uses the HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Windows\AppInit_DLLs registry value [6], which instructs the system to load the DLLs enumerated by its string every time the user32 library initializes into a process. The trick with this value is that it is used even when the operating system boots in Safe Mode, with the exception of *Windows 2003 Server*.

In case AppInit_DLLs is not available the trojan will use an alternate registry key to run at startup by creating a randomly named value containing the string 'rundll32 <rootkit_path>, <export_function>' into one of the following registry subkeys:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

The Windows API provides notification functions for registry subkeys and directories which can signal when the monitored subkeys or directories undertake any changes such as adding, deleting or changing objects and manipulating the security descriptor of any of the contained objects. The functions are FindFirstChangeNotification, FindNextChangeNotification for directories and RegNotifyChangeKeyValue for subkeys.

The rootkit uses a notify function for the directory in which it is installed so that it monitors any changes to its file. If the security descriptor is altered, or if the file attributes are changed or the file is deleted, it will revert any changes made. The same technique is used to monitor the startup registry subkey: if the value is removed or altered while the notification is active it will be restored.

Finally, it uses another registry subkey notification event, this time for HKLM\SYSTEM\CurrentControlSet\Control\Session Manager, in order to monitor the activity of the value 'PendingFileRenameOperations' which is responsible for renaming or deleting files at startup. The purpose of this is that if someone tries to delete the rootkit file using MoveFileEx, which uses that value, it will be able to replace the rootkit file name with random characters, preventing deletion in this way.

CONCLUSION

This isn't the work of an amateur. It may be one of those cases where a virus-writing teenager has refused to face reality and, as a grown up, has gone on to develop and improve his/her evil creatures with a new ultimate purpose: to make a living. It's no longer done for fun. But this is part of evolution and the AV industry must do more in response, since threats such as Linkoptimizer are at large, setting trends in malware development.

REFERENCES

[1] http://www.symantec.com/enterprise/security_response/weblog/2006/10/gromozon_reloaded_everything_t.html.

[2] http://pcalsicuro.phpsoft.it/gromozon.pdf.

[3] http://invisiblethings.org/papers/redpill.html.

[4] http://en.wikipedia.org/wiki/Spaghetti_code.

[5] http://en.wikipedia.org/wiki/Encrypting_File_System.

[6] http://support.microsoft.com/kb/197571.

FEATURE

TROJAN CRIMEWARE – IS IT WORTH ALL THE FUSS?

Jeffrey Aboud
Independent writer, USA

In recent months, the buzz around trojan crimeware has become more prevalent in industry readings. But is the threat real or imagined? Is it something against which we should be building defences, or is it just a marketing ploy, developed in an attempt to sell more security software?

Some of the lack of clarity around these issues may be due, in part, to the variety of labels used to identify this threat. Some security companies and industry organizations refer to these threats generically simply as 'crimeware' or 'cybercrime'; others categorize each type more narrowly, using more specific labels such as 'trojan spyware' or 'trojan malware'.

Regardless of the label used, we define this threat as a malware attack that employs spyware or backdoor trojans. The attack is financially motivated, developed and deployed with the specific intention of stealing sensitive personal information such as online passwords, credit card information, bank credentials, or social security numbers.

MAGNITUDE OF THE PROBLEM

According to a recent report published by *Symantec* [1], five of the top 10 new malicious code families reported during the first six months of 2006 were trojans, and 30 of the top 50 malicious code samples released during that timeframe sought to expose confidential information. Similarly, the national Computer Emergency Response Team for Australia (AusCERT) reports a 46% increase in the total number of identity theft trojan incidents reported between January and October this year, compared with the same period in 2005 (Figure 1).

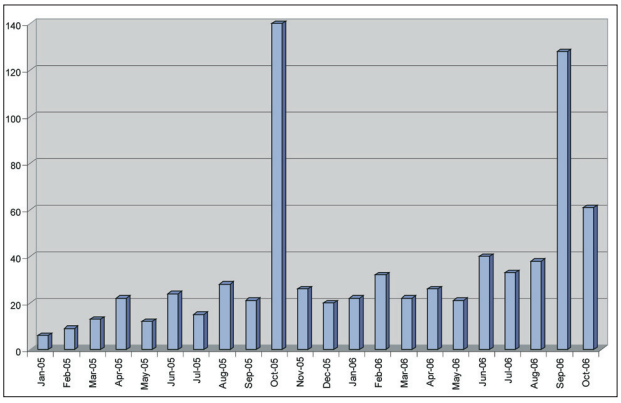


Figure 1: ID theft trojan incidents handled by AusCERT January 2005 to October 2006.

In addition to the rapid growth of trojan spyware and the rest of the crimeware family, the pervasive nature of this threat type, coupled with the relatively high potential for its victims to incur losses, easily justifies the elevation of its status in the threat world.

THE EVOLUTION OF 'INTENT'

Perhaps the single largest change in the security industry in recent years has been the evolution of the intent of malware authors. Whereas 'traditional' viruses were written for the predominant purpose of gaining notoriety, today's bent is definitively in the camp of financial gain.

According to David Perry, global director of education at *Trend Micro*, the number of destructive viruses has historically been small, with the vast majority of viruses doing nothing but replicate. 'We saw the highest number of destructive viruses around 1999–2000, but they've declined steadily since,' explains Perry. 'Nowadays, almost everything we see has a motive.' Mimi Hoang, group product manager for *Symantec Security Response*, agrees. 'The old metric of success for a malware writer used to be who could be first to exploit a particular vulnerability or develop some cool new virus,' says Hoang. 'But today, the metric is based on who can achieve the most comprehensive list of coverage. We see a combination of techniques – such as a series of targeted attacks, coupled with the use of small bot networks – to help spread the infection as far as possible, while staying small enough to remain under the radar.'

This metamorphosis has fuelled an abundance of developments that have become essential components of today's threat landscape. Rather than 'script kiddies', today's malware writers are often skilled programmers, well-versed in a variety of advanced programming techniques; they have the financial wherewithal to obtain formal training and robust software development tools, all of which was out of reach in the recent past. Rather than seeking fame and glory, today's malware authors prefer to go unnoticed, favouring a series of relatively small, targeted infections, rather than a single large-scale attack that attracts a flurry of attention; and instead of individuals who worked on their own with no apparent purpose for their actions, today's threats are increasingly developed by – or with the backing of – organized crime rings, with financial gain as the sole intention behind their actions.

AN ARRAY OF USES

The means by which these threats can achieve their goal is seemingly limitless. Depending on how it is deployed, a

trojan spyware threat can be used to steal anything from passwords, credit card information and the like from consumers, to corporate login credentials from unsuspecting enterprise users, thereby giving the malware owner the keys to the company's most sensitive information.

According to a report published by the Anti-Phishing Working Group (APWG) [2], the ways in which trojan crimeware can be employed to achieve financial gain include:

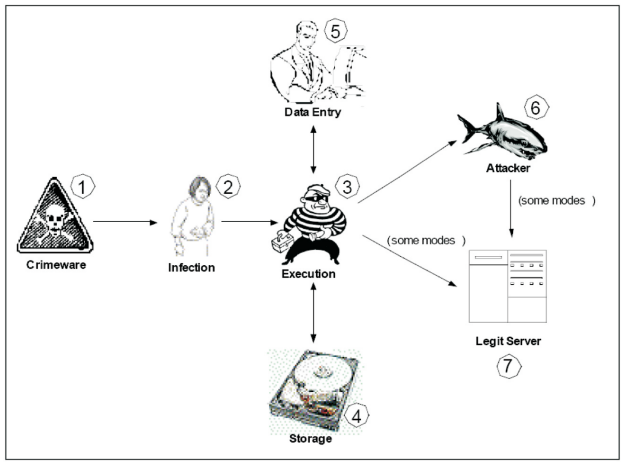
- Theft of personal information.
- Theft of trade secrets and/or intellectual property from businesses.
- Distributed denial-of-service (DDOS) attacks.
- Propagation of spam.
- 'Click fraud' (the simulation of traffic to online advertisements).
- 'Ransomware' (encrypting a user's data, then extorting money from them to have it restored).
- Use of consolidated personal information for furtherance of additional attacks.

Robert Lowe, computer security analyst at AusCERT, adds that most trojan crimeware serves multiple purposes: 'While these trojans have the primary purpose of stealing a range of personal information, including passwords from the compromised computers, these computers are able to be controlled [by the attacker] and used for other purposes to generate financial return, such as supporting further attacks, distributing spam and malware, hosting wares, or taking part in distributed denial of service attacks.'

A HOST OF DISTRIBUTION TECHNIQUES

A significant aspect of the trojan crimeware threat is the variety of distribution techniques that can be utilized to avoid detection. The trojan can be distributed by attaching the code to email and using tried-and-true social engineering techniques to entice users to launch the executable; it can piggyback onto a legitimate application, to be launched simultaneously with the host application; or it can lurk on either a legitimate or sinister website. This list is by no means exhaustive, providing just a flavour for the variety of distribution methods that are at the disposal of this new breed of malware author.

Figure 2 illustrates the transmission methods and corresponding infection points currently used by various types of crimeware. One obvious transmission method that is curiously absent from the list is spam – as are all other forms of mass transmission. Though trojan crimeware can certainly be distributed by mass-mailing techniques, the trend nowadays seems to be that of maintaining a low profile.



Source: APWG

Attack type	Infection point	Data compromise point
Keylogger/screenlogger	2	5 (I/O device)
Email/IM redirector	2	6 (network)
Session hijacker	2	6 (network)
Web trojan	2	5 (I/O device)
Transaction generator	2	N/A
System reconfigurator		
Hostname lookup	3 (execution)	5 (web form)
Proxy	3 (execution)	6 (network)
Data theft	3 (execution - ephemeral)	4 (storage)

Figure 2: Anatomy of a crimeware attack.

As such, most crimeware authors tend to favour a series of small, highly targeted attacks over those released to the masses.

The targeted approach offers the cyber criminal two key benefits. First, by targeting a specific group with similar backgrounds or interests, social engineering techniques can be tailored to appeal specifically toward the commonality of the targeted group, resulting in a higher-than-average chance of success. Second, by keeping the attack small and precise, there is less chance that it will be noticed either by the user or by automated security software.

Once executed and installed, the trojan can remain on the system conceivably forever, waiting for the action that is set to awaken it. And since the longer it remains on the system the higher its likelihood for success, not arousing suspicion is most certainly to the author's advantage.

SUPPORTIVE TECHNOLOGIES

Perhaps more important than the distribution method, however, is the complexity with which many of these threats are written, in an attempt to propagate widely, while still avoiding detection. As mentioned, crimeware authors stand to gain potentially extraordinary financial rewards if they

accomplish their goals. As such, it is well within their interest to make investments in their activities through the purchase of professional tools, training, and the like. And many already enjoy the financial backing of criminal organizations, making it easier to make such investments.

Due to these enhanced capabilities, many of the finished products are quite robust, making their detection and removal non-trivial, at best. For example, many crimeware samples have an overwhelming replication capability: once one process is stopped, the code automatically starts another. Some have the capability to start a large number of simultaneous processes. According to Hoang, during the first six months of 2006, *Symantec* noted trojan code initiating additional processes at an average rate of 11.9 times per day – up from an average of 10.6 times per day in 2005.

Moreover, some observed samples have blended this replication aptitude with polymorphic capabilities, meaning that each time the malware replicates, the ensuing code will be different from the other copies. Since the code will be a slightly different file each time it installs, it is more difficult to detect and remove.

Similarly, the resurgence of rootkit technology over the past 18 months adds yet another layer of complexity to the problem. Over the course of the past year, some crimeware samples discovered in the wild have utilized rootkit technology to mask their processes. Between their cloaking capabilities and their level of system access, rootkits can offer trojan crimeware unparalleled power, as well as the ability to work silently in the background of a system. This new found power can grant crimeware access to more sensitive information, and over a longer period of time – therefore dramatically increasing the likelihood that the user will incur financial losses as a result of the infection.

SECURITY BEST PRACTICES

Though not intended to be an exhaustive list, the following are some guidelines to help protect a company's assets from crimeware:

Keep security definitions updated. Set pattern updates to daily. This is the first line of defence against viruses that can also be hosted on web pages. Many vendors even provide beta definitions with the same quality as the daily download. These should be applied when the threat is severe. Likewise, keep systems patched – particularly systems that are accessible through the corporate firewall.

Make sure the security protection is complete – and overlapping. Regardless of the security provider you use, they should offer a comprehensive, *integrated* suite, which includes multiple layers of protection – from the gateway, to server level, to individual clients. This protection system should be designed by utilizing overlapping and mutually supportive defensive systems. If laptops and other removable devices that can be taken out of the office environment are part of the network configuration, it is essential that the security solution also includes a built-in client-side firewall and an anti-spyware engine, to prevent spyware, backdoors and bots from entering the network when the removable devices are reintroduced. In the age of VoIP, VPN, mobile and cellular devices with network and WiFi capability, border security has been reduced to a myth.

URL filtering. Make sure the anti-virus or anti-spyware product employed has a URL-filtering feature to prevent accidental clicks on known malicious sites. A substantial reduction in this risk can be attained by utilizing IP reputation services, which reside at the gateway.

Educate the users. One of the most basic – yet critical – aspects to protecting a company's resources against a host of threats is to educate users on the need to conduct day-to-day online activity in user-privilege mode, rather than in administrator-privilege mode. This simple difference limits substantially the malicious user's access privileges to system resources in the event of an attack. Additionally, be sure that users are familiar with security best practices, as well as company security policies. It is also prudent to ensure that they understand some of the warnings their security product will provide them, and what these warnings mean (including what actions, if any, they should take when they are presented with such a warning).

Take central control. First, disable any services – inbound or outbound – that are not needed, to limit exposure to only those ports that are necessary to the operation of the business. Second, employ group policies to limit access to critical services, thereby limiting the potential for damage. Third, segregate access points physically and logically and install Network Access Control (NAC) services to ensure all users follow a base model of security. And fourth, configure mail servers to block proactively or remove email attachments with extensions such as .VBS, .BAT, .EXE, .PIF, and .SCR, which are commonly used to spread security threats.

So, is trojan crimeware worth all the fuss? Though the threat is not large enough to make the evening news, it is out there and can cause some real financial losses. Better to build some defences to combat this threat, than to be sorry later.

REFERENCES

[1] Symantec Internet security threat report. September 2006.

[2] Anti-Phishing Working Group (APWG). The crimeware landscape: malware, phishing, identity theft and beyond. October 2006.

COMPARATIVE REVIEW

WINDOWS XP PROFESSIONAL X64 EDITION

John Hawes

64-bit computing is once again the way of the future. After brief flashes of excitement in the 1990s, the *DEC Alpha* and various other proprietary 64-bit systems became confined mostly to specialist use, running their own proprietary UNIX versions, and even the *Intel*/*HP* collaboration the *Itanium* has become something of a niche player.

With the advent of the AMD64 architecture, however, 64-bit has moved out of the server farm and onto the desktop. Only a few years old and rapidly gaining popularity outside the sphere of hardened gamers, speed freaks and other early adopters, machines running on AMD64 (and *Intel*'s version, EM64T) are becoming almost as common as their 32-bit counterparts, with their 32-bit compatibility making the upgrade a fairly painless one. A large part of the long-running row over the security of *Windows Vista*, concerning the *PatchGuard* kernel protection system, applies only to 64-bit platforms, proving the importance of this hardware in the eyes of both operating system and security providers.

A diverse range of products was submitted for this comparative review. Some regulars were notable for their absence – perhaps put off by the platform – while others submitted their standard products hoping that, by virtue of the built-in compatibility, they would work just as well as they do on 32-bit machines. The architecture is still somewhat on the young side however, and oddities of hardware and software are far from uncommon. Beside the usual difficulties associated with testing, I expected the occasional moment of bafflement as the platform, products and tests overlapped in strange new ways. An unusually large number of additions to the In the Wild (ItW) test set also seemed likely to cause a problem or two.

PLATFORM AND TEST SETS

The x86-64 edition of *Microsoft*'s *Windows XP* in fact has rather more in common with *Windows 2003 Server*, and this is immediately obvious from the user experience. Installing to the test lab's suite of 64-bit machines was a simple and remarkably fast process, with the high-powered dual-core AMD64 CPUs, ample RAM and zippy SATA hard drives making light work of the job.

Replicating samples for the *VB* test set was enlivened this month by the arrival of several file infectors in the August WildList, with which the ItW test set was aligned. W32/Detnat, W32/Looked, W32/Virut and W32/Polip, a polymorphic, are all fairly voracious infectors, dropping themselves into opened files or trawling filesystems for likely victims. This allowed several different samples of each to be included in the test set, making a change to the usual worms and bots which have dominated the lists for some time. These, of course, were also represented in some strength, with the expected swathes of W32/Mytob and W32/Areses, along with handfuls of W32/Bagle and other regulars. Most notable among the worms was the advent of W32/Stration, dozens of slightly adapted generations of which continue to be spread worldwide in wave after wave. Most of these I expected to cause little difficulty for the products; the file infectors, on the other hand – particularly the polymorphs – were expected to provide a more probing test of detection capabilities.

Alwil avast! Professional Edition 4.7.902

ItW	100.00%	**Macro**	99.56%
ItW (o/a)	100.00%	**Macro (o/a)**	99.54%
Standard	98.34%	**Polymorphic**	88.22%

The *avast!* product has a resolutely home-user-friendly style about it. The basic GUI has a sleek and sexy appearance, the car-stereo styling providing simple 'Play' and 'Stop' buttons for scanning and a few other basic controls, while a more advanced interface is available for those requiring more fine tuning. This was reached through a small button providing various menu options (which I had ignored at first as it looked like an 'Eject' button, and I assumed it would shut the thing down). The 'Extended' interface provided most of the tools I required, along with a rather bizarre virus information section, featuring a table comparing various aspects of the malware described. While the table clearly showed which items belonged to which sub-grouping, affected which platforms and spread in which ways, the identities of the malware were hidden from the casual browser, and only revealed when an individual line of the table was selected.

With the interface mastered, the product ran along fairly well, although the disabling of scanning certain file types previously scanned by default resulted in several samples being missed (extreme speeds on certain parts of the clean set imply that zip files were among the extensions excluded).

As I have learned from testing *Alwil* products in the past, on-access scanning is not guaranteed to be activated by simple file opening, so some tests required copying test sets to the machine and having the product delete files as they arrived. Eventually *avast!* was cajoled through the tests, missing nothing important and finding nothing but a 'Joke'

On-access tests	ItW File		Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%
Alwil avast! Professional Edition 4.7.902	0	100.00%	21	99.54%	384	88.22%	21	98.95%
Avira Antivir Windows Workstation v.7	0	100.00%	0	100.00%	128	98.13%	0	100.00%
CA eTrust 8.0.403.0	0	100.00%	12	99.82%	103	94.39%	2	99.84%
CAT Quick Heal 2006 v.8.00	0	100.00%	86	97.96%	602	86.05%	97	96.57%
ESET NOD32 v.2.5	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet Forticlient 3.0.349	8	99.86%	0	100.00%	15	99.86%	0	100.00%
GDATA AntiVirusKit 2007 v.17.0.6282	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG 7.5.427	0	100.00%	0	100.00%	249	91.88%	22	98.60%
Kapsersky Anti-Virus 6.0.0.303	0	100.00%	0	100.00%	0	100.00%	2	99.69%
McAfee VirusScan Enterprise 8.0i	0	100.00%	0	100.00%	46	97.14%	0	100.00%
Norman Virus Control v.5.82	3	99.90%	0	100.00%	309	91.01%	6	99.59%
Sophos Anti-Virus 6.0.5	0	100.00%	8	99.80%	0	100.00%	15	99.30%
Symantec Antivirus 10.1.5.5000	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend Micro OfficeScan Corporate Edition 7.3	1	99.95%	13	99.68%	851	92.64%	30	98.67%
VirusBuster VirusBuster Professional 2006 (x86-64) v.6.0	0	100.00%	8	99.80%	123	93.90%	25	99.12%

in the clean set, therefore becoming the first product to receive a VB 100% award this month.

Avira Antivir Windows Workstation v.7

ItW	100.00%	**Macro**	99.93%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	97.50%

Avira's now-familiar shiny, happy style led me through a simple installation, past a warning to ensure I had a genuine copy of the software rather than a cheap rip-off, into the equally straightforward interface. Controls were where I expected to find them (perhaps through some familiarity with the product as much as judicious design), and the little umbrella in the system tray marking the status of the on-access protection opened and closed smoothly and quickly as I adjusted the settings for various tests.

Scanning speeds were fairly decent, and most of the collections were handled pretty thoroughly, with a smattering of zoo samples missed but nothing in the ItW set. In the clean set, the false positive spotted last time around has long since been fixed, so there was nothing to deny *Avira* a VB 100% this time.

CA eTrust 8.0.403.0

ItW	100.00%	**Macro**	99.82%
ItW (o/a)	100.00%	**Macro (o/a)**	99.82%
Standard	99.96%	**Polymorphic**	98.16%

CA's *eTrust* product has been submitted in more or less the same form throughout my experience here at *VB*; with a new version looming, this could be the last appearance of this incarnation on the test bench. The large corporate installer, with its numerous EULAs, lengthy activation code and sizeable page of personal information to fill out, including access passwords for the configuration controls, took longer than most despite familiarity. As usual, I opted to install the agent parts only, without any of the extra network management tools, and after some time setting up was faced with the browser-based GUI. The testing itself also dragged over some time, with the GUI taking its time to respond when trying to switch between tabs. Displaying of logs was particularly drawn out; at one point, bored of watching the progress display telling me my logs would be ready to view in a moment, I wandered off to grab a drink, only to find on my return that my 'session' had timed out. Revisiting the logging tab and repeating the

process, I was again distracted by other things, overestimating the length of the 'session' and finding myself once more back at the start.

In terms of scanning itself, things were quite different. Awesome speeds were achieved, both in the clean set and over infected areas, with detection pretty decent throughout – suggesting the engine, if not the interface, was making efficient use of the powerful hardware. The old *InoculateIT* engine, not used by default and therefore not eligible for the VB 100%, displayed some even quicker scanning speeds over some of the test sets, although detection was not as thorough as the *Vet* engine and some strange anomalies popped up when trying this option (including, for a brief moment, a file in the clean set locked by the on-access scanner – an event which could not be reproduced). With no false positives to report from the *Vet* engine, and little missed elsewhere, *eTrust* wins itself a VB 100%.

CAT Quick Heal 2006 v.8.00

ItW	100.00%	**Macro**	98.23%
ItW (o/a)	100.00%	**Macro (o/a)**	97.96%
Standard	96.57%	**Polymorphic**	86.05%

The *Quick Heal* installation process included a quick scan of 'system areas' to ensure it was safe to install to my machine. After the setup and a reboot, a friendly message welcomed me to the product, and led me into the main GUI, a sharp and crisp affair with the shadowy image of a masked face barely visible in the background. The clean and simple controls hid no surprises, apart from a rather cute bug-in-gun-sights motif which seemed a little out of place amongst the seriousness shown elsewhere.

The generally well-designed interface did leave something to be desired when I couldn't figure out how to disable the pop-ups warning of on-access detections. A vast swathe of these overwhelmed my machine on one attempt, but eventually the on-access test was coaxed to completion. On demand, the product more than lived up to its name, zipping merrily through speed tests and virus collections, although OLE2 processing was not as impressive as other file types, and detection of some of the more obscure entries in the zoo collections was less than perfect. With nothing missed from the ItW test set though, *Quick Heal* earns a VB 100%.

ESET NOD32 v.2.5

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

ESET's product had its usual fast and simple installation experience, sprinkled with green-tinged *Matrix*-style graphics and, at one point, a rather scary-looking eye I hadn't spotted on previous tests. Also along the way was an option to connect to *ESET*'s *ThreatSense* system, to submit samples of detected malware to its researchers, and also the choice of whether or not to activate the on-access scanner by default on startup. Declining both of these, I played around with the GUI, having fun with separable and reconnectable panes, dragging them around the screen in various configurations only to be a little disappointed by the more standard *XP*-style of the main scanner. Now familiar with the rather obscure naming system of its modular functions, I found my way around easily, and the product powered through the tests with its usual highly impressive combination of speed and accuracy.

A few wobbles occurred, although my main annoyance, a momentary lingering after quitting from a scan job, would have seemed less noticeable on a product that ran at normal speed. A strange message shown on deactivating some monitors, telling me they would be completely uninstalled on reboot, seemed to have no lasting effect. With splendid and remarkably consistent speed, and irreproachable detection, *NOD32* takes another VB 100% award in its stride.

Fortinet Forticlient 3.0.349

ItW	99.86%	**Macro**	100.00%
ItW (o/a)	99.86%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	99.86%

FortiClient has a somewhat sombre feel; its installation is fast and efficient and its interface grey and simple, light on graphics and heavy on text. The multi-tabbed controls left little to be desired, being easy to navigate and pretty comprehensive, giving me no problems in carrying out the tests. Speeds were very good over OLE2 files, though no more than decent elsewhere, and detection was pleasantly strong across the zoo sets. Just when all seemed to have gone well, checking the logs of the ItW test set showed that an entire variant of one of the newly added file infectors, W32/Looked, was not spotted, either on access or on demand, putting paid to *FortiClient*'s chances of a VB 100% award.

GDATA AntiVirusKit 2007 v.17.0.6282

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

On-demand tests	ItW File		Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	Number missed	%	Number missed	%
Alwil avast! Professional Edition 4.7.902	0	100.00%	18	99.56%	384	88.22%	33	98.34%
Avira Antivir Windows Workstation v.7	0	100.00%	3	99.93%	131	97.50%	0	100.00%
CA eTrust 8.0.403.0	0	100.00%	12	99.82%	85	98.16%	1	99.96%
CAT Quick Heal 2006 v.8.00	0	100.00%	73	98.23%	602	86.05%	97	96.57%
ESET NOD32 v.2.5	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Fortinet Forticlient 3.0.349	8	99.86%	0	100.00%	15	99.86%	0	100.00%
GDATA AntiVirusKit 2007 v.17.0.6282	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Grisoft AVG 7.5.427	0	100.00%	0	100.00%	249	91.88%	19	98.74%
Kapsersky Anti-Virus 6.0.0.303	0	100.00%	0	100.00%	0	100.00%	0	100.00%
McAfee VirusScan Enterprise 8.0i	0	100.00%	0	100.00%	46	97.14%	0	100.00%
Norman Virus Control v.5.82	3	99.90%	0	100.00%	309	91.01%	4	99.71%
Sophos Anti-Virus 6.0.5	0	100.00%	8	99.80%	0	100.00%	15	99.30%
Symantec Antivirus 10.1.5.5000	0	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend Micro OfficeScan Corporate Edition 7.3	1	99.95%	13	99.68%	851	92.64%	30	98.67%
VirusBuster VirusBuster Professional 2006 (x86-64) v.6.0	0	100.00%	8	99.80%	123	93.90%	21	99.45%

Next year's version of *AntiVirusKit* looked as futuristic as its title, with slick and shiny design and graphics, including the red-and-white shield logo, shimmering and glittering from the screen. After the zippy install and a reboot, the GUI itself was just as shiny and funky, with the usual clearly laid out controls given a zing and a fizz of colour. Setup was simple and straightforward, with the option to drop 'Engine A' or 'Engine B' ignored in favour of the default double-barrelled approach. As expected, this scanning style did not produce record times in the speed tests, but accuracy was beyond reproach, with only a 'Joke' in the clean set requiring me to make any further entries in my test notes. *GDATA* now has another VB 100% award for its trophy cabinet.

Grisoft AVG 7.5.427

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	98.74%	**Polymorphic**	91.88%

Compared to its neighbours on the test bench, *Grisoft*'s product looked positively dour, its greyish install process enlivened only by the rather useful option to create a rescue disk. The interface itself was also drab and grey and serious and, like many products aimed more firmly at the home user market, used the approach of providing a basic interface for the general user and an advanced one for those who require more specific settings. Tinkering away in here provided me with most of the configuration tools I needed to get through my tests, although when it came to saving logs I had some difficulty, and dumped numerous listings of the on-screen options to file before I discovered that the simpler interface was the way to go. Getting the results of my scans all on one screen enabled me to save them to file, and parsing showed solid detection, along with reasonable if unremarkable speeds. Missing nothing significant, and entirely without false positives, *AVG* also earns itself a VB 100% award.

Kaspersky Anti-Virus 6.0.0.303

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

The *Kaspersky* interface for this product forms a major part of the company's *Internet Security Suite*, which I reviewed

in some depth for these pages a few months ago (see *VB*, September 2006, p.16), so I expected to have no difficulties with it. With my brain swamped by so many AV products in recent months, it took me a few moments to refresh my acquaintance with the large, fist-friendly GUI, but had it doing my bidding in no time. Installation was very fast, with no reboot required, and testing passed in similarly painless fashion, running over the sets in respectable time and getting the expected impressive results. With the only samples missed being on-access, in file types not scanned by default in that mode, *Kaspersky 6* is another worthy recipient of the VB 100% award.

McAfee VirusScan Enterprise 8.0i

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	97.14%

McAfee's *VirusScan* product, after 'recomposing' its constituent parts in a rather leisurely fashion prior to install, thanked me politely for making use of it as it set itself up. Once installed, the product was its usual unfussy self, its bare GUI and straightforward layout allowing for fairly simple adjustment of the appropriate options. Tests proceeded without problems, at a decent pace and with reliable detection, the product proving to be more than good enough to earn a VB 100%.

Norman Virus Control v.5.82

ItW	99.90%	**Macro**	100.00%
ItW (o/a)	99.90%	**Macro (o/a)**	100.00%
Standard	99.71%	**Polymorphic**	91.01%

Norman's product also has a multi-window approach, with various functionality provided by separate areas, but here it seemed somewhat disjointed, with some desired options falling between the gaps. The installation was simple enough, with the friendly green traffic-light man leading the way. Setup, configuration and running of scans was done via various control systems, with some options set globally and others as part of the scan 'task'. Running a scan, a separate window carried the results and hid away in a minimized state if nothing was found, quietly slipping away again at the end if the user didn't demand to see it.
On-access testing was equally fiddly, with unpredictable behaviour forcing me to resort to deletion. Scans were a little slow over some sets, but remarkably fast over OLE2 files, and detection rates were pleasantly regular in both on-access and on-demand tests. Unfortunately this consistency extended to the missing of three samples of W32/Detnat, added to the WildList used for this round of testing, thus denying *Norman* a VB 100% award.

Sophos Anti-Virus 6.0.5

ItW	100.00%	**Macro**	99.80%
ItW (o/a)	100.00%	**Macro (o/a)**	99.80%
Standard	99.30%	**Polymorphic**	100.00%

Installation of *Sophos Anti-Virus* was fast and simple, and using the product was equally unchallenging – until the point at which the result logs needed collecting. Configuration of this functionality seems limited in the end-user interface, perhaps moved to some higher level of the administration suite, but these issues were soon circumvented and useable logs acquired (although one *Linux* server I passed them to for parsing insisted they were in MPEG format). My only complaint apart from this was the progress bar, always more of an art than a science, which here seemed to either rush to 95% and hang around there for some time, or to complete the scan with the bar still on 10%. With its usual solid detection rates, *Sophos* also receives the VB 100% award.

Symantec Antivirus 10.1.5.5000

ItW	100.00%	**Macro**	100.00%
ItW (o/a)	100.00%	**Macro (o/a)**	100.00%
Standard	100.00%	**Polymorphic**	100.00%

Symantec's product was almost ruled out of the game at a very early stage, when the supplied version announced it was not compatible with my processor, and a standby 32-bit version, spotting my swanky hardware, instructed me to install the 'Win64' product which had just brushed me off. On consultation, it emerged that an *Itanium* product had been provided in error, and I was pointed to the more appropriate AMD64 version, which ran without further difficulty. This product differed little at the user end from its counterparts, and setup and running of the tests was simple and rapid.

Scanning speed was decent, if not remarkable, over the clean sets, but a repeat of last month's issues of extreme slowdown over the infected collections threatened to upset things once more, especially as the deadline for this review drew rapidly closer. However, the problem had been

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables			Zipped OLE Files			Dynamic files		
	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	FPs [susp]
Alwil avast! Professional Edition 4.7.902	47.6	13522.0		3.6	22037.2		2.0	79708.3		1.0	74607.5		7.6	6347.7	
Avira Antivir Windows Workstation v.7	95.3	6751.8		3.7	21675.9	1	3.7	43556.4		5.0	14921.5		37.7	1281.0	
CA eTrust 8.0.403.0	24.0	26818.7		3.0	26444.6		34.0	4688.7		6.7	11202.3		2.7	18136.3	
CAT Quick Heal 2006 v.8.00	23.7	27204.1		10.7	7442.2		23.0	6931.2		11.0	6782.5		14.7	3290.8	
ESET NOD32 v.2.5	16.0	40228.0		2.0	39666.9		2.0	79708.3		1.0	74607.5		1.0	48242.6	
Fortinet Forticlient 3.0.349	173.3	3713.4		4.0	19833.4		75.7	2107.0		3.7	20384.6		6.7	7243.6	
GDATA AntiVirusKit 2007 v.17.0.6282	153.0	4206.9		25.3	3132.0		67.3	2367.7		27.0	2763.2		23.7	2039.0	
Grisoft AVG 7.5.427	75.9	8484.7		6.1	12941.9		29.6	5385.7		6.4	11603.0		11.4	4231.8	
Kapsersky Anti-Virus 6.0.0.303	107.7	5978.5		7.0	11333.4		21.7	7360.0		8.0	9325.9		6.7	7243.6	
McAfee VirusScan Enterprise 8.0i	48.0	13409.3		6.0	13222.3		18.7	8543.2		3.3	22404.7		10.0	4824.3	
Norman Virus Control v.5.82	385.0	1671.8		3.0	26444.6		69.7	2288.5		3.0	24869.2		50.7	952.3	
Sophos Anti-Virus 6.0.5	39.7	16229.2		7.0	11333.4		11.7	13672.1		4.0	18651.9		10.0	4824.3	
Symantec Antivirus 10.1.5.5000	64.0	10057.0		4.3	18321.9		27.0	5904.3		5.3	13997.7		4.7	10352.5	
Trend Micro OfficeScan Corporate Edition 7.3	32.3	19908.7		2.0	39666.9		13.7	11670.3		2.0	37303.7		5.7	8523.4	
VirusBuster VirusBuster Professional 2006 (x86-64) v.6.0	91.3	7047.5		2.3	34048.8		61.3	2599.3		9.7	7723.3		17.7	2731.7	

diagnosed by *Symantec* techs as 'non file-related scanning', and a supplied utility to counter the effects of this got me my collection results at an impressive rate. Detection was even more impressive, and *Symantec* joins those at the top of the podium, not putting a foot wrong anywhere and earning its VB 100% award with ease.

Trend Micro OfficeScan Corporate Edition 7.3

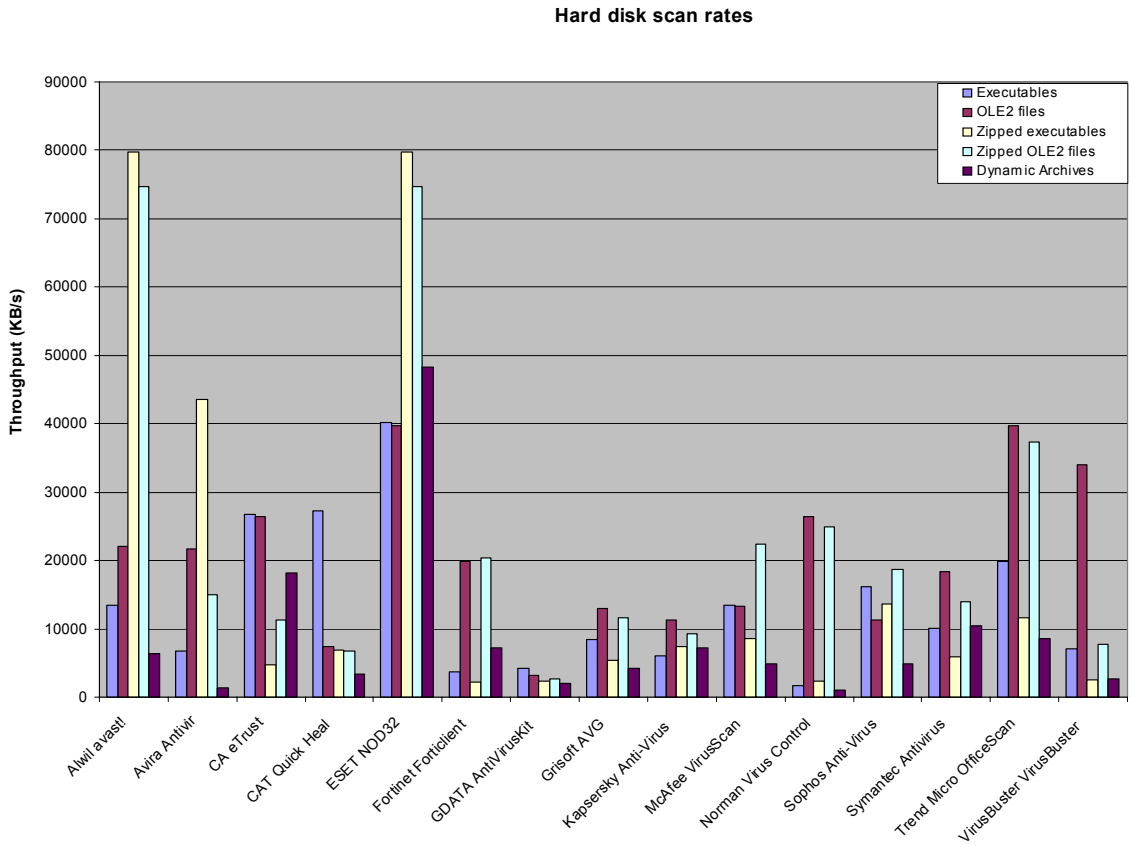
ItW	99.95%	**Macro**	99.68%
ItW (o/a)	99.95%	**Macro (o/a)**	99.68%
Standard	98.67%	**Polymorphic**	92.64%

Nearing the end of my set of products, and the time allotted to my testing, *Trend* also presented me with 64-bit-related difficulties. When run on one of the machines set up for this review, the product seemed at first to have frozen during the installation, until switching windows revealed a message box lounging behind the drab green of the installer backdrop, informing me that the product could not be installed on my system. Checking with contacts at *Trend*, I learned that the 64-bit version could not be installed directly, but must be deployed via the management system, only available for 32-bit hardware. With time ticking by, I hurriedly set up a second machine with a *Windows 2000* image from the previous comparative, installed the server product (which entailed, as in the earlier test, upgrading my browser), and from there was able to 'Notify' the client of the availability of a product. This installed via http, with half a dozen messages from the *XP* security system querying whether I really wanted to install, but with those dealt with I finally had a serviceable scanner.

Much of the administration was also carried out via the server, including changes to on-access settings and access to logs. Speed of scanning was very good, and after a few anomalous sets of results were cleared up by retesting, detection was fairly decent too, though a few sizeable sets of older polymorphic viruses were missed. More importantly, a single sample of W32/Detnat was not spotted in the WildList set, in either mode, spoiling the product's chances of an award.

VirusBuster VirusBuster Professional 2006 (x86-64) v.6.0

ItW	100.00%	**Macro**	99.80%
ItW (o/a)	100.00%	**Macro (o/a)**	99.80%
Standard	99.45%	**Polymorphic**	93.90%



VirusBuster, last on the test bench, provided a 64-bit version of its product, but its looks and operation were more or less indistinguishable from other editions. The installation process presented various standard options, including where to install the product and whether to set up a desktop shortcut, before I could 'actualize the anti-virus protection.' I found the layout of the GUI somewhat fiddly, requiring a fairly lengthy process of designing scan tasks and then running them. The product had another rather misleading progress bar, often starting off at around 80%, and took a long time writing out its logs when asked to, but had no trouble with detection and got through the speed tests at a decent rate. Once again, some somewhat flaky results meant a second run over the tests was needed, but in the end *VirusBuster* proved itself capable of handling the ItW set without problems, and so also earns a VB100%.

CONCLUSIONS

As expected, the test produced some upsets, with the new file-infector viruses causing trouble for several products. With few misses of ItW viruses over the first few months of my tenure here at *VB*, this proved a bumper crop, with three products failing to cover the whole list accurately, and one missing an entire variant – others missed only some samples, while detecting others spawned from the same source. False positives were less of a problem, after some cleaning out of the clean set, and overall coverage of the zoo collections has also improved almost across the board, since little new material was added for this test. The expected platform issues were limited to some confusion from vendors over which products to submit, and how they could be installed, and were soon overcome with a little investigation and advice from the providers.

Some considerable redesign of the VB 100% testing setup and processes is due, hopefully in time for the next comparative in two months' time. More details will be made available nearer to the time.

Technical details: All tests were run on identical AMD Athlon 64 3800+ dual core machines with 1GB RAM, 40GB and 200 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive, running *Microsoft Windows XP Professional x64* edition.

Virus test sets: Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/Win64/2006/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

CALL FOR PAPERS

VB2007 VIENNA

Virus Bulletin is seeking submissions from those wishing to present papers at VB2007, which will take place 19–21 September 2007 at the Hilton Vienna, Austria.

The conference will include a programme of 40-minute presentations running in two concurrent streams: Technical and Corporate. Submissions are invited on all subjects relevant to anti-malware and anti-spam.

In particular, *VB* welcomes the submission of papers that will provide delegates with ideas, advice and/or practical techniques, and encourages presentations that include practical demonstrations of techniques or new technologies.

SUGGESTED TOPICS

The following is a list of topics suggested by the attendees of VB2006. Please note that this list is not exhaustive – the selection committee will consider papers on any subjects relevant to the anti-malware community.

- In-line scanning
- Malware on mobile platforms
- Demonstrations of malware in action
- Rootkits
- Cross-device malware
- Advanced disinfection and prevention techniques
- Law enforcement – tales from the trenches, cooperation between anti-malware industry and law
- Emulation, unpacking techniques
- Behavioural detection
- Anti-malware testing
- *Vista* security issues
- Mac OSX malware
- Unix malware
- Shellcode
- Anti-malware market analysis and statistics
- Reverse engineering
- Network forensics
- Hardware virtualization
- Application proxies
- Corporate case studies
- Spyware and adware
- Defence in depth
- Image spam
- Spam filter performance testing
- Latest anti-spam techniques
- Use of spam filters in the corporate environment
- Proactive defence against phishing
- Convergence of spam and virus solutions
- Motivation of malware writers
- Machine learning for malware detection
- 64-bit threats
- Botnets – analysis, case studies
- Automating malware analysis
- IM threats
- VoIP threats
- Polymorphism
- Malware on console games
- Data acquisition for corpus building
- AV backscatter and abuse reporting
- IDS/IPS
- Corporate budgeting for security
- Malware classification
- Detection of compiled malware

HOW TO SUBMIT A PROPOSAL

Abstracts of approximately 200 words must be sent as plain text files to editor@virusbtn.com no later than **Thursday 1 March 2007**. Submissions received after this date will not be considered. Please include full contact details with each submission.

Following the close of the call for papers all submissions will be anonymized before being reviewed by a selection committee; authors will be notified of the status of their paper by email. Authors are advised that, should their paper be selected for the conference programme, the deadline for submission of the completed papers will be Monday 4 June 2007. Full details of the paper submission process are available at http://www.virusbtn.com/conference/.

NEW FOR 2007

In addition to the traditional 40-minute presentations, *VB* plans to trial a new concept at VB2007. A portion of the technical stream will be set aside for a number of 20-minute, 'last-minute' technical presentations, proposals for which need not be submitted until two weeks before the start of the conference. This will encourage presentations dealing with up-to-the-minute specialist topics. There will be no limit on the number of proposals submitted/presented by any individual, and presenting a full paper will not preclude an individual from being selected to present a 'last-minute' presentation. Further details will be released in due course.

END NOTES & NEWS

AVAR 2006 will be held 4–5 December 2006 in Auckland, New Zealand. For full details, conference agenda and online registration see http://www.aavar.org/.

The 22nd ACSAC (Applied Computer Security Associates' Annual Computer Security Conference) takes place 11–15 December 2006 in Miami Beach, FL, USA. Alongside a technical program and a 'work in progress session' attendees may also register for a workshop on host-based security assessment and tutorials on subjects that include biometric authentication, malware, live forensics, security engineering, next-generation wireless risks, certification and accreditation, and large-scale network traffic analysis. For details see http://www.acsac.org/.

The 2nd AVIEN Virtual Conference will take place online on Wednesday 10 January 2007, from 16:00 to 18:00 GMT (starting at 8am PST, 11am EST). This year's conference topic is 'The new face of malware: stories from the battlefield'. Sign-up details will be announced in due course.

RSA Conference 2007 takes place 5–9 February 2007 in San Francisco, CA, USA. The theme for this year's conference – the influence of 15th century Renaissance man Leon Battista Alberti, the creator of the polyalphabetic cipher – will be covered in 19 conference tracks. For full details see http://www.rsaconference.com/2007/US/.

Black Hat Federal Briefings & Training 2007 take place 26 February to 1 March 1 2007 in Arlington, VA, USA. Registration for the event will close on 18 February 2007. For details see http://www.blackhat.com/.

Websec 2007 will take place 26–30 March 2007 in London, UK. More information will be available in due course at http://www.mistieurope.com/.

Black Hat Europe 2007 Briefings & Training will be held 27–30 March 2007 in Amsterdam, the Netherlands. Early (discounted) registration closes 12 January. For online registration and details of how to submit a paper see http://www.blackhat.com/.

The 16th annual EICAR conference will be held 5–8 May 2007 in Budapest, Hungary. A call for papers for the conference has been issued with a deadline of 12 January 2007 for peer-reviewed papers. Full details can be found at http://conference.eicar.org/.

The 22nd IFIP TC-11 International Information Security Conference takes place 14–16 May 2007 in Sandton, South Africa. Papers offering research contributions focusing on security, privacy and trust are solicited. For more details see http://www.sbs.co.za/ifipsec2007/.

The 8th National Information Security Conference (NISC 8) will be held 16–18 May 2007 at the Fairmont St Andrews, Scotland. For the conference agenda and a booking form see http://www.nisc.org.uk/.

The 19th FIRST Global Computer Security Network conference takes place 17–22 June 2007 in Seville, Spain. For full details see http://www.first.org/conference/2007/.

The International Conference on Human Aspects of Information Security & Assurance will be held 10–12 July 2007 in Plymouth, UK. The conference will focus on information security issues that relate to people. For more details, including a call for papers, see http://www.haisa.org/.

Black Hat USA 2007 Briefings & Training takes place 28 July to 2 August 2007 in Las Vegas, NV, USA. Registration will open on 15 February. All paying delegates also receive free admission to the DEFCON 15 conference. See http://www.blackhat.com/.

The 17th Virus Bulletin International Conference, VB2007, takes place 19–21 September 2007 in Vienna, Austria. See p.21 for the call for papers. Online registration and further details will be available soon at http://www.virusbtn.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues):

- Single user: $175
- Corporate (turnover < $10 million): $500
- Corporate (turnover < $100 million): $1,000
- Corporate (turnover > $100 million): $2,000
- *Bona fide* charities and educational institutions: $175
- Public libraries and government organizations: $500

Corporate rates include a licence for intranet publication.

See http://www.virusbtn.com/virusbulletin/subscriptions/ for subscription terms and conditions.

Editorial enquiries, subscription enquiries, orders and payments:
Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England
Tel: +44 (0)1235 555139 Fax: +44 (0)1235 531889
Email: editorial@virusbtn.com Web: http://www.virusbtn.com/

vb Spam supplement

CONTENTS

NEWS & EVENTS

ANTI-SPAMMER LOSES CASE

An anti-spam activist has successfully been sued in a US federal court by the company he accused of spamming.

VB reported on Mumma's story last year (see *VB*, April 2005, p.S1). After receiving four unsolicited emails from online travel agent *Cruise.com*, Mumma lodged a complaint with the travel agent's parent company *Omega World Travel*. When he continued to receive emails from *Cruise.com*, Mumma documented this fact, along with the history of the complaint, on his website 'sueaspammer.com'. Unhappy with being labelled a spammer, *Omega* filed a lawsuit arguing that Mumma had violated its trademark and copyright by using images of the company's founders and that he had defamed individuals associated with *Cruise.com* with the comments posted on his site.

Unfortunately for Mumma, the US Court of Appeals for the Fourth Circuit sided with the alleged spammers.

Technology law blogger Eric Goldman explains the Court's reasoning in his blog (http://blog.ericgoldman.org/). According to Goldman, the 'falsity and deception' exclusion to CAN-SPAM's pre-emption only covers fraud or other types of tortious misrepresentation, which means that state anti-spam laws cannot be used to pursue spammers for 'immaterial falsity or deception'. And since a significant number of states have re-enacted their anti-spam laws since the introduction of CAN-SPAM, relying on the 'falsity and deception' standard, it is expected that this ruling will cast significant doubt on the enforceability of most of those state laws.

The judges also declared that, while the header information contained in the alleged spam messages was false, and the 'from' address was non-functional, these 'mistakes' were immaterial because the emails were 'chock full' of other methods to identify, locate, or respond to the sender.

Finally, the court said that Mumma had 'failed to submit any evidence that the receipt of 11 commercial email messages placed a meaningful burden on [his] company's computer systems or even its other resources.'

Goldman says 'Unquestionably, the defendants benefited from having made a reasonably good faith effort to comply with CAN-SPAM, even if they didn't dot every i and cross every t. But even if they tried to be good actors, they are still allegedly spammers, which makes this result an amazing hat trick for the defendants – no liability under the Oklahoma state anti-spam law, CAN-SPAM or common law trespass to chattels. As Fourth Circuit precedent, surely this opinion will take some wind out of the sails of anti-spam plaintiffs.'

EVENTS

A workshop on countering spam will be held on 8 December 2006 as part of the ITU Telecom World 2006 event in Hong Kong. The workshop will present the activities of relevant organizations and consider the potential for future cooperative measures and partnerships for countering spam. For details see http://www.itu.int/WORLD2006/.

The 9th general meeting of the Messaging Anti-Abuse Working Group (MAAWG) will take place 29–31 January 2007 in San Francisco, CA, USA. Members and non-members are welcome. Registration closes on 25 January. Two further general meetings will also take place in 2007: 5–7 June in Dublin, Ireland (members only) and 3–5 October in Washington D.C. (open to all). For details see http://www.maawg.org/.

The 2007 Spam Conference is tentatively scheduled to take place on 30 March 2007 at MIT, Cambridge, MA, USA. The proposed title for this year's conference is 'Spam, phishing and other cybercrimes'. See http://spamconference.org/.

The Authentication Summit 2007 will be held 18–19 April 2007 in Boston, MA, USA. The two-day intensive program will focus on online authentication, identity and reputation, highlighting best practices in email, web and domain authentication. Presentation proposals are currently being reviewed, with a submission deadline of 15 December 2006. For full details see http://www.aotalliance.org/summit2007/.

Inbox 2007 will be held 31 May to 1 June 2007 in San Jose, CA, USA. For more details see http://www.inboxevent.com/.

FEATURE

THE MEDIUM OR THE MESSAGE? DEALING WITH IMAGE SPAM

Catalin Alexandru Cosoi
BitDefender, Romania

Internet users have recently become used to a new category of spam message: one in which the content is delivered in the form of an image attachment. A year ago, such 'image spam' accounted for approximately 10% of the total amount of spam circulating. In recent months, however, spammers have noticed that many of the current anti-spam solutions are almost ineffective against this trick so they have started attacking this niche in earnest. Image spam has increased to 30–40% of the total amount of circulating spam, with the addition of random noise making almost every image unique. Detection rates have dropped even further.

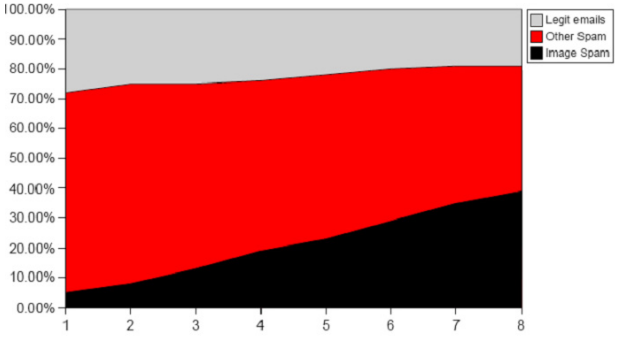


Figure 1: The evolution of image spam during the months March – October 2006. (Months labelled 1 – 8.)

Performing any sort of content analysis on such emails requires an Optical Character Recognition (OCR) module. Yet common OCR filters are computationally expensive and their accuracy leaves much to be desired.

An alternative approach would be to use a filter that ignores the text contained within the image (i.e. the message, from a human point of view) and instead learns by experience some common characteristics of the images themselves (the chosen medium or vector of communication), thus achieving reliable detection. With this idea in mind, our research concentrated on finding a way to represent and subsequently detect specific characteristics of spam images.

We were looking for a comparison function that is permissive enough to ignore noise, but sufficiently discriminative to avoid false positives, while still demanding reasonable amounts of processing power.

For this purpose, we took a new look at histogram extraction and comparison (a histogram can be defined in this case as a list of colours and their relative preponderance in an image; it tells us what colours exist in an image and how many pixels are of a given colour).

These are, of course, tried and tested techniques for content-based image retrieval (CBIR) that are fast enough to be run on almost any modern system, and also discriminating enough to provide a pretty good detection rate.

However, the false positive rate of such techniques is rather high. Of course, when you're just searching for pretty pictures, more *is* better. In an anti-spam solution, on the other hand, false positives are a serious problem. The types of distance functions commonly met in old CBIR systems are: histogram Euclidean distance, histogram intersection distance and histogram quadratic (cross) distance. Of all these options, the most appealing for an anti-spam engine would be the histogram intersection distance.

Experimentation revealed that this is potentially useful because it can ignore changes in background colours, but problematic when changes appear in the foreground. Also, if the noise added by the spammers uses the same colours but in a different quantity, these colours will add to the distance.

With a view to alleviating the problems of the classical algorithm, we introduced a new type of histogram distance, designed specifically for spam images – let's call it Spam Image Distance, or SID.

Equation 1 shows the definition of SID, where a, b and c are the quantities of red, green and blue, and h(a, b, c) represents the number of pixels occupied by the colour created by mixing a% red, b% green and c% blue.

In other words, we consider not only identical colours, but also similar ones, which can differ from the original by no more than δ values, with the restriction that we consider this element in the sum to be non-zero only if the difference between the sizes of the bins is smaller than Δ. These two parameters can be determined using Receiver Operating Characteristic (ROC) curves (simultaneous comparison of sensitivity and specificity), trial and error, or by using a machine-learning technique.

While this new technique can be shown to perform well on 'clean' images, there remains the problem of noise

$$SID(h,g) = \frac{\left(\sum_{A} \sum_{B} \sum_{C} \min\left(h(a,b,c), g(a\pm\delta, b\pm\delta, c\pm\delta)\right) \left| \frac{h(a,b,c) - g(a\pm\delta, b\pm\delta, c\pm\delta)}{\min\left(|h(a,b,c)|, |g(a\pm\delta, b\pm\delta, c\pm\delta)|\right)} \times 100 \le \Delta \right)}{\min\left(|h|, |g|\right)}$$

Equation 1.

elimination. Fortunately, the techniques used by spammers to add noise or otherwise obfuscate the images are well known to us.

Common 'noising' techniques catalogued at this time include:

a. Adding random pixels to the image.

b. Animated GIFs with noisy bogus frames.

c. Similar colours between different parts of the text in the image.

d. A long line at the end of the image (some kind of border) with random parts missing.

e. Splitting the image into sub images and using the table facilities in HTML to reconstruct the image.

f. Sending different sizes of the same image.

g. Image poisoning – inserting legitimate image content such as company logos into spam messages.

The arsenal of countermeasures is similarly wide. For instance, to eliminate random pixel noise from an image histogram we can use this simple function:

$$H' = \left\{ c_i \in H \left| \frac{|c_i|}{\sum_{j=1}^{|H|} |c_j|} \times 100 > \gamma \right. \right\}$$

and let SID deal with the outcome. Another useful trick is to 'stitch together' the histograms of images embedded in HTML tables (if they are sufficiently similar) and then let SID consider the resulting composite histogram.

The distances found by the SID function are used to compare images that are already in the spam database with images that we want to add. If the image analysis returns a score smaller than a threshold T, then we add the image. Otherwise, we consider it to be a known image. The SID can only fail if an image is entirely new or if it is a malformed image from which we can't extract a histogram. In our current experience, new images appear at the rate of one per day.

Some care should be applied to deciding exactly what the filter learns, as spammers have started using company logos as noise in spam images. Misidentifying a company logo as a spam image could create a serious problem.

Some other sources of false positives exist as well. Using a filter that compares histograms does not tell one anything about the content of the pictures (the colours of human skin, for instance, are the same no matter whether the picture is explicit in nature or just an innocuous vacation snapshot).

DETECTION RATES

When run against the *BitDefender* corpus of spam images (a few million samples extracted from real spam) SID shows a 98.7% detection rate. Within the corpus 1.23% of images are malformed and we can't extract the histograms for those pieces of spam, but this is not considered to be a significant problem since the image cannot be seen by the user either.

A further 0.03% represent false positive results. If we delete from the corpus all those images that are malformed, the detection rate quickly jumps to 100%.

We believe that SID is a worthwhile addition to the arsenal of any modern spam hunter and that advances in noise reduction will further improve the potential of this already very useful tool.

BIBLIOGRAPHY

[1] Sablak, S.; Boult, T. E. Multilevel color histogram representation of color images by peaks for omni-camera. Proceedings of IASTED International Conference on Signal and Image Processing, 1999. See http://vast.uccs.edu/~tboult/PAPERS/IASTED99-Multilevel-color-histogram-representation-of-color-images-by-peaks-for-omni-camera--Sablak-Boult.pdf.

[2] Stevens, M.R.; Culberston, B.; Malzbender T. A histogram-based color consistency test for voxel coloring. Proceedings of 16th International Conference on Pattern Recognition, 2002. See http://www.hpl.hp.com/research/mmsl/publications/vision/icpr02-final.pdf.

[3] Scheunders, P. A comparison of clustering algorithms applied to color image quantization. Pattern Recognition Letters Vol. 18, No. 11, 1997. See http://webhost.ua.ac.be/visielab/papers/scheun/prl97.pdf.

[4] Pass, G.; Zabih R. Histogram refinement for content-based image retrieval. Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision, 1996.

[5] Ling, H.; Okada, K. Diffusion distance for histogram comparison. IEEE Conference on Computer Vision and Pattern Recognition, 2006. See http://www.cs.umd.edu/~hbling/Research/Publication/336_ling_h.pdf.

[6] Graham-Cumming, J. The rise and rise of image-based spam. Virus Bulletin Spam Supplement. November 2006 p. S2. See http://www.virusbtn.com/sba/2006/11/sb200611-image.